AA203
Optimal and Learning-based Control

Adaptive and learning MPC

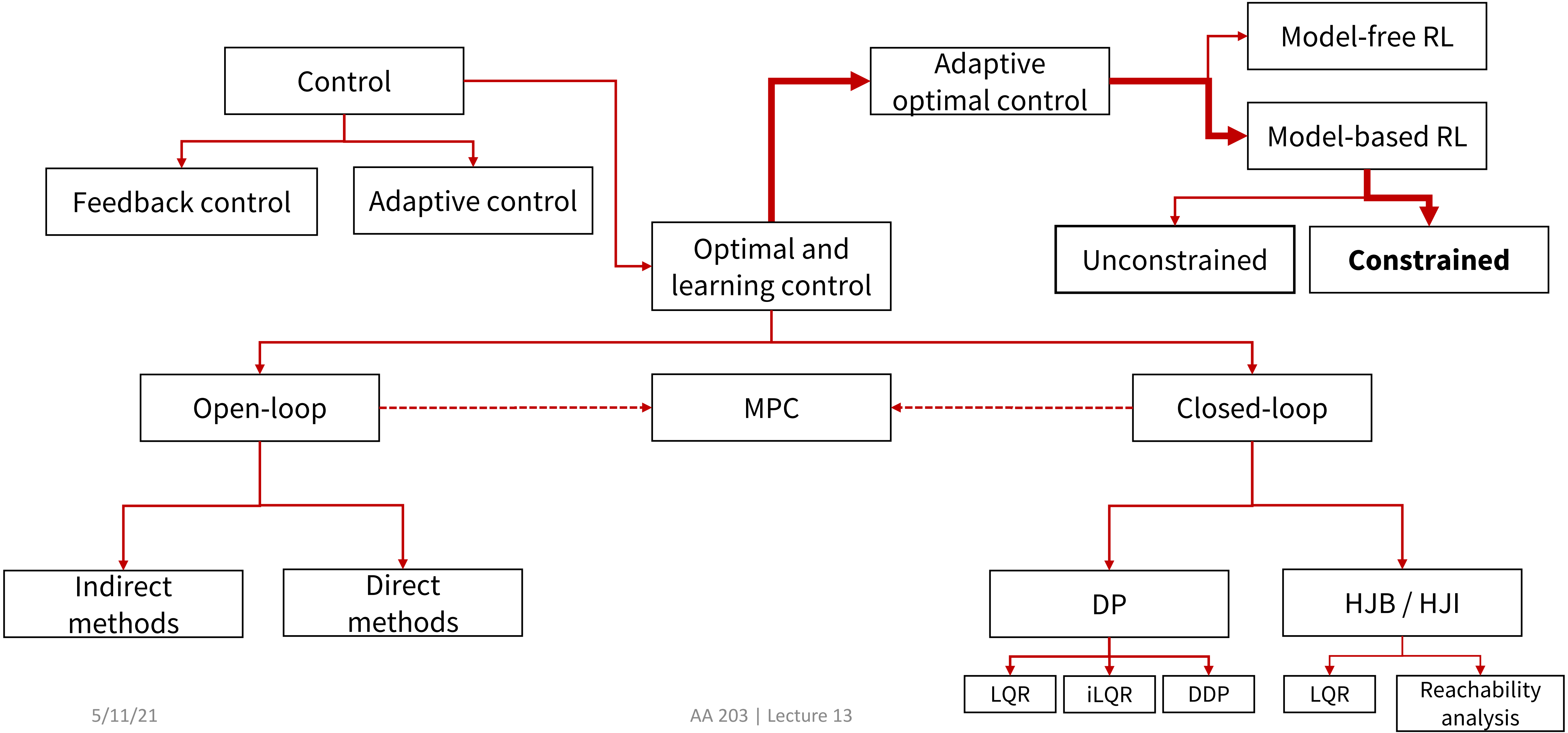
Stanford University

Autonomous Systems Lab

Logistics

- Project reports being graded now
- 2/3 term survey going out on Wednesday, due Sunday
- Homework 3 released today, due Wednesday the 26th

Roadmap


Control
Feedback control
Adaptive control
Optimal and learning control
Adaptive optimal control
Model-free RL
Model-based RL
Unconstrained
Constrained
Open-loop
MPC
Closed-loop
Indirect methods
Direct methods
DP
HJB / HJI
LQR
iLQR
DDP
LQR
Reachability analysis

Adaptive and Learning MPC

- Learning MPC as an example of learning/adaptive constrained control
- Practical considerations
- Learning quantities other than dynamics

- Reading:
  - L. Hewing, K. P. Wabersich, M. Menner, M. N. Zeilinger. *Learning-Based Model Predictive Control: Toward Safe Learning in Control.* Annual Review of Control, Robotics, and Autonomous Systems, 2020.
  - U. Rosolia, X. Zhang, F. Borrelli. *Data-Driven Predictive Control for Autonomous Systems.* Annual Review of Control, Robotics, and Autonomous Systems, 2018.

Learning dynamics

- Approach:
  - Learn dynamics and maintain a measure of uncertainty
  - Incorporate uncertainty into controller to guarantee constraint satisfaction
    - Using e.g. robust MPC
- Model learning types:
  - Robust/Set-membership models
    - Typically easier analysis, potentially sensitive to misspecification
  - Statistical models (e.g. least squares estimation)
    - More difficult analysis

Robust estimation models

- Setting: given operation data

$$X = [\mathbf{x}(0), \dots, \mathbf{x}(K+1)], \qquad U = [\mathbf{u}(0), \dots, \mathbf{u}(K)]$$

from system

$$\mathbf{x}(t+1) = f(\mathbf{x}(t), \mathbf{u}(t), \mathbf{w}(t), \boldsymbol{\theta})$$
$$\mathbf{w}(t) \in W \ \ \forall t$$

- Approach: maintain feasible parameter set

$$\mathrm{T}_K = \{\boldsymbol{\theta}: \forall t = 0, \dots, K \ \exists \mathbf{w} \in W \ \text{ s.t. } \mathbf{x}(t+1) = f(\mathbf{x}(t), \mathbf{u}(t), \mathbf{w}(t), \boldsymbol{\theta})\}$$

Set of *non-falsified* parameters

Robust estimation models

- Note that $T_{K+1} \subseteq T_K$: once a parameter value is falsified, it is removed from the feasible set forever.
- Frequently used consequence:
  - Let $U = [\mathbf{u}(0), \ldots, \mathbf{u}(N)]$ denote a feasible open loop action sequence from state $\mathbf{x}(0)$ for all $\boldsymbol{\theta} \in T_K$. Then, U is feasible for all $\boldsymbol{\theta} \in T_{K+n}$ with $n \geq 0$ (from the same state $\mathbf{x}(0)$).

Additive linear example

- Dynamics
$$\mathbf{x}(t+1) = A\mathbf{x}(t) + B\mathbf{u}(t) + \mathrm{E}\boldsymbol{\theta} + \mathbf{w}(t); \quad \mathbf{w}(t) \in W$$

E known, $\boldsymbol{\theta}$ unknown.

- Assume initial polytopic uncertainty T_0.
- Polytopic constraints $F\mathbf{x} \leq \mathbf{f}, G\mathbf{u} \leq \mathbf{g}$.

Additive linear example

Bujarbaruah, Zhang, Rosolia, Borrelli. *Adaptive MPC for Iterative Tasks*, CDC 2019.

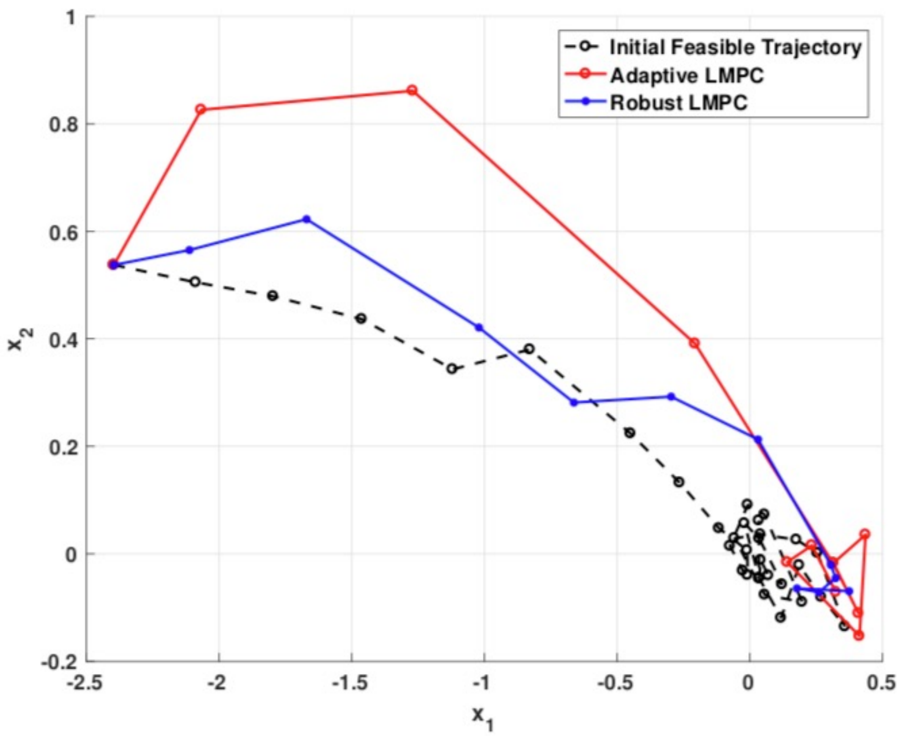
- Let X_0 denote terminal invariant associated with dynamics and T_0.
- Then, X_0 also invariant for $T_n, \mathrm{n} \geq 0$.
- Approach: At timestep n, consider combined disturbance

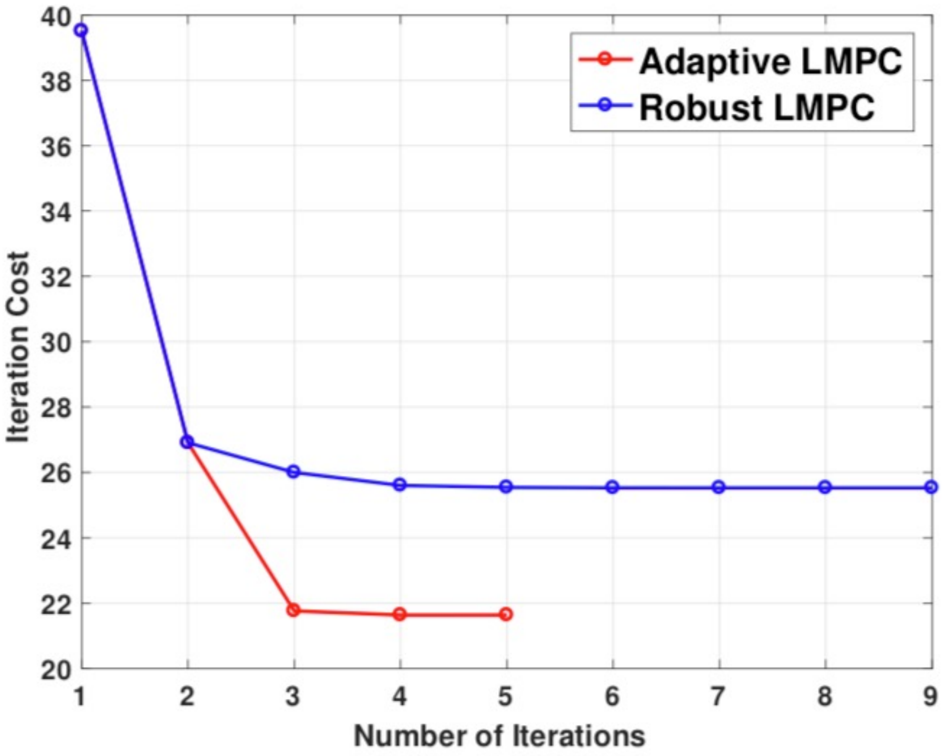
$$\mathbf{d}(t) = E\boldsymbol{\theta} + \mathbf{w}(t), \qquad \boldsymbol{\theta} \in T_n$$

Use robust/tube MPC to solve.

- Can also adapt terminal invariant, will see later.

Additive linear example





Robust MPC

- Many similar approaches for
  - Multiplicative uncertainty
    $$\mathbf{x}(t+1) = A\mathbf{x}(t) + B\mathbf{u}(t) + \mathbf{w}(t), (A, B) \text{ unknown.}$$
  - Nonlinear (but linearly parameterized) uncertainty
    $$\mathbf{x}(t+1) = A\mathbf{x}(t) + B\mathbf{u}(t) + \Phi\big(\mathbf{x}(t), \mathbf{u}(t)\big)\boldsymbol{\theta}$$
- Also exist robust *non-parametric* methods

Decoupling safety and performance

A. Aswani, H. Gonzalez, S. S. Sastry, C. J. Tomlin. *Provably safe and robust learning-based model predictive control*. Automatica, 2013.

- We have so far considered learning a model and using this model for performance.
- Instead consider safety model

$$\mathbf{z}(t+1) = A\mathbf{z}(t) + B\mathbf{u}(t) + \mathbf{w}(t); \quad \mathbf{w}(t) \in W$$

where W is assumed known, and performance model

$$\mathbf{x}(t+1) = A\mathbf{x}(t) + B\mathbf{u}(t) + g(\mathbf{x}(t), \mathbf{u}(t))$$

- Optimize cost for $\mathbf{x}(t)$ subject to polytopic constraints on $\mathbf{z}(t)$.

Stochastic estimation models

System

$$\mathbf{x}(t+1) = f(\mathbf{x}(t), \mathbf{u}(t), \mathbf{w}(t), \boldsymbol{\theta})$$

with $\mathbf{w}(t) \sim p(\mathbf{w})$ iid.

Common assumption: noise appears linearly

$$\mathbf{x}(t+1) = f(\mathbf{x}(t), \mathbf{u}(t), \boldsymbol{\theta}) + \mathbf{w}(t)$$

Approach:

- Use tools from *probabilistic* estimation (e.g. max likelihood, Bayesian inference, etc)
- Construct confidence intervals or credible regions to *probabilistically* guarantee safety

Confidence sets

- In set-membership identification, we constructed sets that contained the parameters with probability 1
- In this section, we will consider sets of the form $T_k(\delta)$ such that
$$p(\,\boldsymbol{\theta} \in T_k(\delta) \mid X_k, U_k\,) \geq 1 - \delta$$
- Similarly, can no longer reason about constraints being satisfied with probability 1, must work with *chance constraints*

Chance-constrained optimal control problem

$$J_0^*(\mathbf{x}_0) = \min_{\mathbf{u}_0,\ldots,\mathbf{u}_{T-1}} p(\mathbf{x}_T) + \sum_{k=0}^{T-1} c(\mathbf{x}_k, \mathbf{u}_k)$$

subject to $\mathbf{x}_{k+1} = A\mathbf{x}_k + B\mathbf{u}_k + \mathbf{w}_k, \quad k = 0, \ldots, N-1$

$\mathbf{w}_k \sim p(\mathbf{w})$ iid, $k = 0, \ldots, N-1$

$p(\mathbf{x}_k \in X \; \forall k) \geq 1 - \delta_x$

$p(\mathbf{u}_k \in U \; \forall k) \geq 1 - \delta_u$

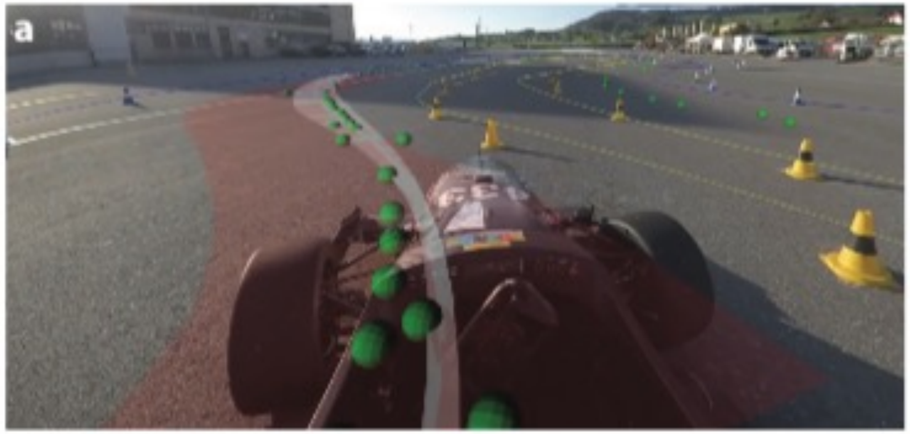
Computing confidence sets

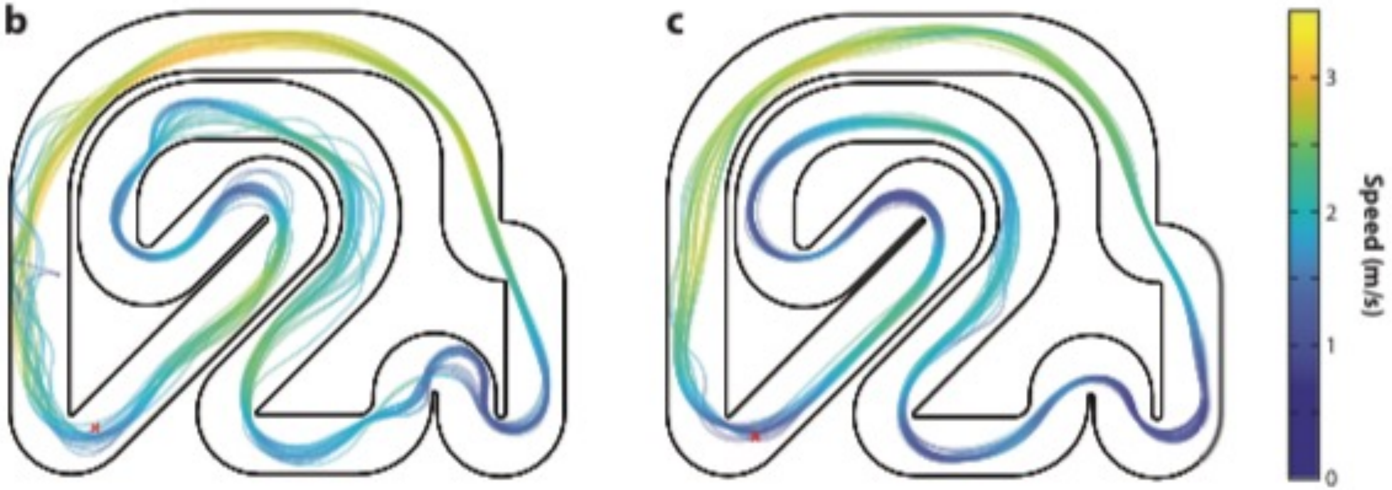
- Most common approach: take Bayesian approach, assume noises is Gaussian
  - Model: linearly parameterized or Gaussian process
- Frequentist approaches:
  - Bootstrapping
  - If noise model sub-Gaussian, can use concentration inequalities (effectively yields same result as Gaussian confidence intervals)

A robust approach to stochastic control

- Simple set-theoretic computations of robust MPC are convenient
- Common approach: divide “risk” equally over timesteps, so at each time constraints must be satisfied with probability $1 - \delta/T$
- Then guarantee that $T_K(\frac{1-\delta}{T})$ satisfied constraints; better chance constraint satisfaction typically relies on Monte Carlo methods
- Typically over-conservative in practice
- Recursive feasibility arguments difficult

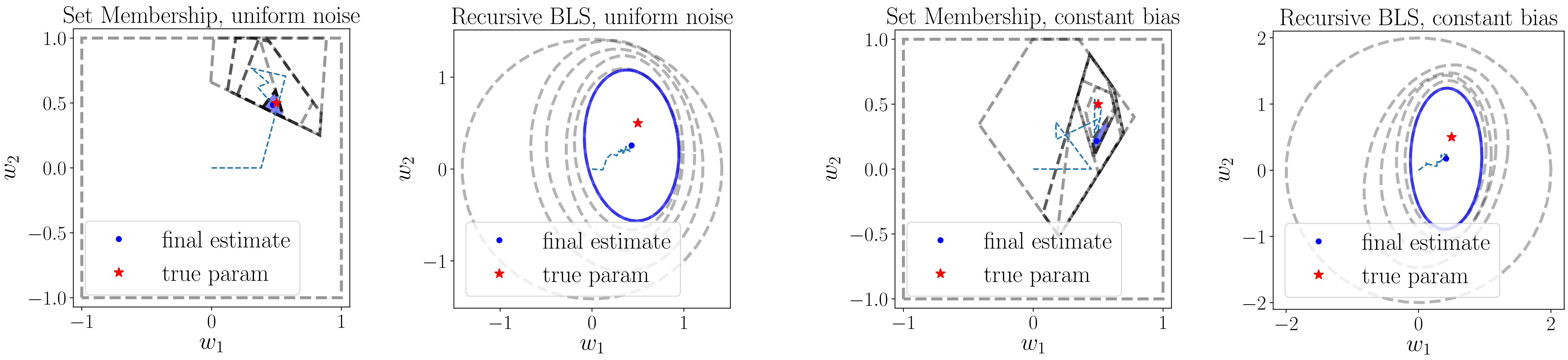
Application





Estimator comparison

- System: $y(t) = w_1\, \phi_1(x(t)) + w_2\, \phi_2(x(t)) + v(t)$



Sinha, Harrison, Richards, Pavone. Under review.

Learning the terminal constraint

- Line of work from Rosolia and Borrelli over multiple papers (2017-2020)
- Assume we have access to terminal control invariant X_f.
- Know that backward reachable set of X_f is also invariant
- Therefore, given trajectory $\{\mathbf{x}(0), \dots, \mathbf{x}(N+1)\}$ such that $x(N+1) \in X_f$, know:

$$X_f \cup \{\mathbf{x}(0), \dots, \mathbf{x}(N)\}$$

is control invariant.

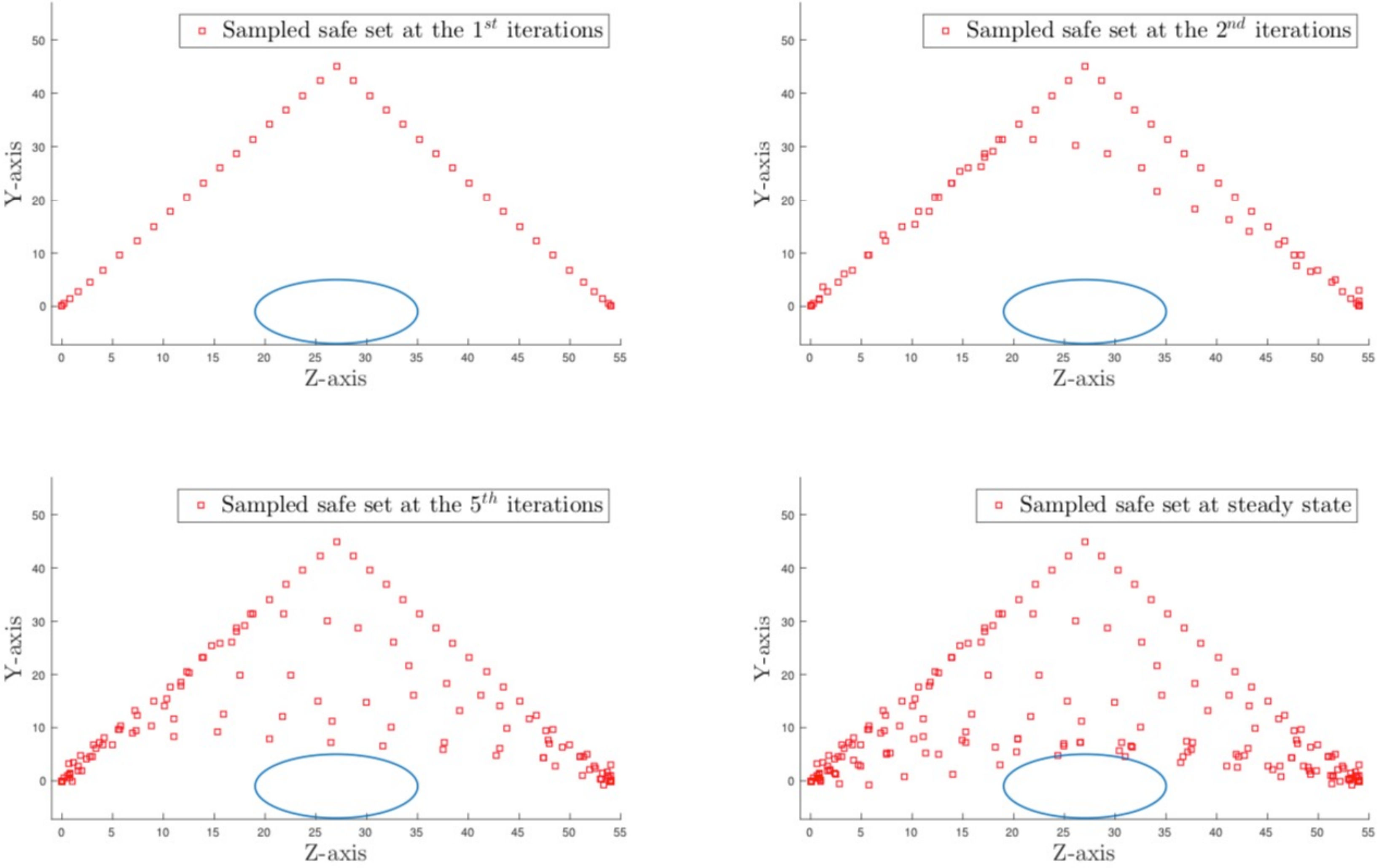
Learning the terminal constraint

- Algorithm: assume access to a demonstration trajectory or stabilizing controller.
- Initialize $X_f = \{0\}$
- Iterate over episodes k = 1,…
  - Each episode k yields data
    $$D_k = \{\mathbf{x}_k(0), \dots, \mathbf{x}_k(N)\}, \;\; C_k = \{c(\mathbf{x}_k(0)), \dots, c(\mathbf{x}_k(N))\}$$
  - Expand terminal constraint via
    $$X_f \leftarrow X_f \cup D_k$$
  - Terminal cost $p(\mathbf{x})$ is the sum of all future costs from the last time that state was visited
  - Solve MPC problem with terminal constraint X_f and terminal cost $p(\mathbf{x})$

Learning the terminal constraint

- Can show that for systems without disturbances, this results in monotonic performance improvement.
- In practice, to make optimization problem tractable, use convex hull of sampled set and weighted sum of tail costs.
- Blanchini & Pellegrino (2005) showed that the convex hull of the sampled set is also control invariant for LTI systems!

Performance



Iteration	Iteration Cost
$j = 0$	65.000000000000000
$j = 1$	33.634529488066327
$j = 2$	24.216166714512450
$j = 3$	19.625000000001727
$j = 4$	19.625000000000004
$j = 5$	17.625000000022546
$j = 6$	17.625000000000000
$j = 7$	16.625000000000000
$j = 8$	16.625000000000000

Learning the terminal cost

- Important to also learn the terminal cost.
- Simple approach: use the tail cost from the previous visit to a given state

What else could we learn?

- Learn terminal cost: use e.g. similar ideas to Q-learning
- Learn controller hyperparameters (e.g. planning horizon)
- Learn constraints (based on e.g. binary signals of constraint violation)
- Learning from demonstrations (behavioral cloning, imitation learning–not covered in this class but practically very useful)

Next time

- Unconstrained model-based methods in the tabular and nonlinear setting